94

[redacted]icroprocessors

est 1

100 Points MAX

21 Points

1. **Number Representation:** In the MIPS processor, we can represent numbers in either signed or unsigned format. Assuming that our word size is **7 bits**, fill the blanks in the table below using **both** binary values **and** their decimal equivalent.

$\omega = 7$

Number	Unsigned	Signed
Maximum value which can be represented	127 = ~~1~~111 1111 ✓	63 = ~~0~~011 1111
Minimum value which can be represented.	0 = ~~0~~000 0000 ✓	−64 = ~~1~~100 0000 ✓
17	17 = ~~0~~001 0001 ✓	17 = ~~0~~0010001 ✓
-5	**N/A**	−5 = 1111 0011 ✗

− 15

UNSIG: MAX → $2^{\omega}-1 \Rightarrow 2^{7}-1 = 128-1=127$

MIN → 0

SIG: MAX → $2^{\omega-1}-1 \Rightarrow 2^{6}-1 = 64-1=63$

MIN → $-2^{\omega-1} \Rightarrow -2^{6} = -64$

```
1111 1010
        1
---------
1111 1011
```

Take 1's Comp FIRST – then add!

32
16
––
48
12
––
60

```
 7   6  5  4    3 2 1 0
 0   0  0  0    0 0 0 0
128 64 32 16    8 4 2 1
```

```
      0000 0101
−5 ⇒       0001
      ---------
      0000 1100 → FLIP BITS
```

4 Points

2. Given the following binary value, 101101010, what is the equivalent hexadecimal value?

1 : 6 : A

0x16A

− 4

________0xB50

___X____0x16A

________0x552

________0x362

19

CISC	RISC
Few GPR	Many GPR
Many Addr.	Few Addr.
One - many	fixed.

6 Points

3. Given a processor which is a RISC processor, select the characteristics below which apply:

_________Few general-purpose registers — CISC.

____X____Few addressing modes ✓

____X____Many general-purpose registers

____X____Fixed length instructions

_________Instructions require one to many clock cycles to execute CISC.

6

4. In the MIPS processor that is part of the PIC32 microcontroller, what is the length of each instruction? **4 Points**

______ 8 bits

______ 16 bits

___X___ 32 bits ✓

______ 64 bits

4

5. In the MIPS processor, how many bits are used to represent a byte and how many are used to represent a half-word? **4 Points**

8 bits ⇒ 1 Byte.

16 bits ⇒ Half-Word ⇒ 2 Bytes.

4

6. What are the three basic operations in processing an instruction in a stored program computer? **6 Points**

Fetch, Decode & Execute

6

20

7. In the PIC microcontroller, we use SFRs. What are SFRs used for and how do they differ from general purpose registers? **_10 Points_**

SFR's communicate & configure peripherals. ✓ 10

GPR'S store data & addresses.

8. The MIPS processor has a load-store instruction set. Explain what is meant by load-store. **_10 Points_**

You load data from MEMORY INTO ~~a~~ register, ~~but~~ before you can~~not~~ manipulate it ~~yet~~. Therefore, there is ~~no~~ memory to memory operations. ✓ 6

after→ store back TO MEMORY

9. If our MIPS code has the following instruction: **_10 Points_**

lw $t1, 8($t0) where $t0 contains 0xB00C

	Memory contents	Address	
		Address	
-12	0XCD99	0xB000	
-8	0xA100	0xB004	
-4	0x4888	0xB008	
$t0→	0x6541	0xB00C	
4	0x722B	0xB010	
8	0x4220	0xB014	←
12	0xCA0A	0xB018	
16	0x1BB7	0xB01C	
20	0x2000	0xB020	← WANT
24	0x78B0	0xB024	

9

a. Which value would be written into $t1?

0x4220 ✓

b. What instruction would you use to write the value in $t1 to memory address 0xB020?

~~lw~~ $t1, 20($t0)

SW

25

10 Points

10. In the MIPS assembly language code shown for the MinMax routine, explain in detail what will happen if a nop instruction is not included after the following instruction, which appears on line 16:

ble $t0,$v1, chk

```
1   .text       # Pipelined Implementation
2   MinMax:
3               lw      $v0, 0($a0)
4               addiu   $a0, $a0, 4
5               addi    $a1, $a1, -1
6               blez    $a1, ret
7               move    $v1, $v0
8   loop:
9               lw      $t0, 0($a0)
10              addi    $a0, $a0, 4
11              bge     $t0, $v0, next
12              nop
13              b       chk
14              move    $v0, $t0
15  next:
16              ble     $t0, $v1, chk
17              nop
18              move    $v1, $t0
19  chk:
20              addi    $a1, $a1, -1
21              bnez    $a1, loop
22              nop
23  ret:
24              jr      $ra
25              nop
```

BRANCH

^ would happen if this nop was removed.

5

We would have a Control Hazard if a nop wasn't included b/c we are dealing with a branch function. If it was not included, ~~it would "branch" to the chk function & do addi $a1, $a1, -1.~~

The move func would always be exec. & we would not have our min/max function properly.

So what happens?

11. When we are configuring all of the bits in a ports as inputs, we must also configure the open-drain setting for each of them. **5 Points**

5

____ TRUE

_X_ FALSE ✓

OUTPUTS → OPEN-DRAIN

INPUTS → PULL-UP.

10

$2^{10} = 1024$

$2^6 = 64$

$$\begin{array}{r} {}^{1\,2} \\ 1024 \\ 64 \\ \hline 4096 \\ 61440 \end{array}$$

65536

12. You are using timer 1 to measure a specific time interval. With your configuration of the clock source and the maximum pre-scaler option, you determine that you will need to count 67,322 clock pulses. This is possible using timer 1. (Explain why, or why not.) **5 Points**

5

____ YES

X NO

$2^{16} - 1 = 65,536 - 1 = \underline{65,535}$

✓ We cannot use timer 1 as it only has 65,535 alloted clk pulses, and we need 67,322.

13. Our microcontroller uses a system clock for controlling the processor operations (SYSCLK). We use a separate clock to drive the peripheral devices (PBCLK). Why do we use different clocks and how are they typically configured with respect to each other? **5 Points**

5

SYSCLK → 80MHZ

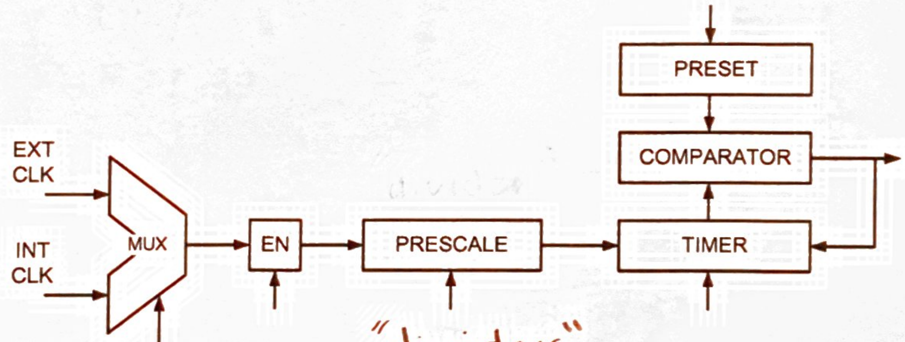
USBCLK → 48MHZ

PBCLK → 10MHZ

✓ We wouldn't use all the clks at the same time b/c of their different speeds. We use them in specific scenarios. ✓

14. Shown below is the block diagram for a timer. Explain what the prescaler does and why we might want to use different configurations. **10 Points**

10



"divider"

The pre-scaler, or divider, helps us verify our timer can handle the clk pulses, similar to what we did in prob. 12. ✓ We'd use diff. configs to find the most efficient clk & timer in our machine.

20